

Sähköinen rakennusasiakirja



Ammattikorkeakoulun opinnäytetyö

Tietotekniikan koulutusohjelma

Riihimäki, syksy 2016

Joona Korkki



Riihimäki
Tietotekniikan koulutusohjelma
Web-kehitys

Tekijä	Joona Korkki	Vuosi 2016
Työn nimi	Sähköinen rakennusasiakirja	

Opinnäyte perustuu rakennusyritykselle tuotettuun pilvipalveluun. Projektin aiheena oli luoda sähköinen versio rakennustarkastukseen liittyvistä dokumenteista. Pääasiallinen tavoite oli siis saada jo olemassa olevat dokumentit sähköiseen muotoon muokattavaksi mobiililaitteilla. Projektin lopputuloksen olisi oltava kokonaisuudessaan helpompi käyttää kuin vanha paperidokumentteihin perustuva työskentely.

Työ käsittelee pilvipalvelun toteuttamista AWS-ympäristössä sekä sovelluksen toiminnan ja ulkoasun luomista. Aiheessa sivutaan käytettyjä web-tekniikoita ja alustoja kuten Bootstrap ja UserFrosting.

Avainsanat Pilvipalvelu, HTML5, CSS3, PHP, AWS

Kohderyhmät Tietotekniikan insinöörit ja muut verkkoasiantuntijat

Sivut 23 s.

Riihimäki
Degree Programme in Information Technology
Web-development

Author	Joona Korkki	Year 2016
Subject of Bachelor's thesis	Electronic construction document	

This thesis is based on a cloud service that was created for a construction company. The subject of the project was to create an electronic version of building inspection documents. The main objective was to get existing documents into an electronic format for editing on mobile devices. As a whole the final product should be easier to use than the old way of using paper documents.

The work deals with the implementation of a cloud service in the AWS environment as well as the creation of the applications behavior and appearance. The topic briefly involves used web technologies and platforms such as Bootstrap and UserFrosting.

Keywords Cloud service, HTML5, CSS3, PHP, AWS

Target audience IT engineers and other web-experts

Pages 23 p.

LYHENTEET JA TERMIT

W3C	World Wide Web Consortium
WWW	World Wide Web
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
Tagi	Tag
PHP	PHP: Hypertext Preprocessor
Backend	Data access layer
Frontend	Presentation layer
AJAX	Asynchronous JavaScript and XML
JSON	JavaScript Object Notation
CRUD	Create, Read, Update, Delete
AMI	Amazon Machine Image
EBS	Elastic Block Store
EC2	Amazon Elastic Compute Cloud
API	Application programming interface
Volyyimi	Logical drive

SISÄLLYS

1	JOHDANTO.....	1
2	PALVELIMEN TOTEUTUS.....	1
2.1	Katsaus käytettyihin web-tekniikoihin.....	1
2.1.1	HTML5.....	1
2.1.2	CSS3.....	2
2.1.3	JavaScript.....	2
2.1.4	jQuery.....	2
2.1.5	PHP.....	3
2.2	Amazon EC2.....	3
2.2.1	EC2-instanssin luominen.....	3
2.3	LAMP-palvelin.....	5
2.3.1	Linux.....	6
2.3.2	Apache.....	6
3	KÄYTTÄJIEN HALLINTAJÄRJESTELMÄ.....	7
3.1	Käyttöönotto.....	7
3.2	Uuden käyttäjän luominen.....	9
3.3	Uuden sivun liittäminen navigaatiopalkkiin.....	10
3.4	Turvallisuus.....	11
3.4.1	Salasanojen turvaaminen.....	12
3.4.2	Cross-site request forgery (CSRF).....	12
3.4.3	Cross-site scripting (XSS).....	12
3.4.4	SQL-injektio.....	12
4	PILVIPALVELUN TOTEUTUS.....	13
4.1	Toiminnallisuus.....	13
4.2	Käyttöliittymä.....	14
4.3	Kuvien ottaminen.....	18
4.4	Dokumenttien tulostus.....	18
4.5	Varmuuskopiointi.....	19
4.6	Laajennusmahdollisuudet.....	20
4.6.1	UserFrosting päivitys.....	20
4.6.2	Synkronointi tiedostopalveluun.....	20
4.6.3	Sähköpostipalvelimen lisääminen.....	21
4.6.4	LAMP-palvelimen päivitys.....	21
5	YHTEENVETO.....	21
	LÄHTEET.....	22

1 JOHDANTO

Projektin asiakas on erikoistunut pientalon vastaavan työnjohtajan tehtäviin ja halusi pilvipalvelun tukemaan normaalia työskentelyään. Valmis palvelu korvasi rakennustyömaalla täytettävien tarkastusasiakirjojen paperiversion kokonaan sähköisellä versiolla.

Työ sisältää LAMP-palvelimen pystyttämisen Amazonin tarjoamia palveluita käyttäen ja palvelimen varmuuskopiointiin liittyvät asiat sekä itse palvelun luomisen. Palvelua käytettäisiin mobiililaitteilla työmaakäyntien aikana joten mobiiliystävällisten verkkosivujen suunnittelu oli aina pidettävä mielessä.

Opinnäytetyön aikana tutustuttiin myös UserFrosting-nimisen käyttäjänhallintajärjestelmän käyttöön ja sen ominaisuuksiin. Järjestelmään yhdistettiin myös REST-ohjelmointirajapinta palvelun toimintojen luomiseksi. Lopuksi pohditaan projektin onnistumista kokonaisuutena.

2 PALVELIMEN TOTEUTUS

Palvelimen toteuttamisessa otettiin huomioon mikä ratkaisu sopii parhaiten asiakkaalle. Pilvipalvelu sopii hyvin pienemmille yrityksille joilla harvoin löytyy tarvetta ja henkilöstöä kiinteälle palvelimelle ja sen ominaisuuksille.

2.1 Katsaus käytettyihin web-tekniikoihin

Tärkeimpinä niistä tekniikoista, joita käytetään verkkosivuja- ja sovelluksia tehtäessä mainittakoon HTML5, CSS3 ja JavaScript. Verkkosivut tukeutuvat vahvasti näiden kolmen muodostamaan pohjaan. Yhtälöön voidaan kuitenkin lisätä eri koodikirjastoja kuten jQuery tai palvelinpuolen ohjelmointikieliä kuten PHP luomaan vielä lisää toiminnallisuutta sivuille. Tekniikoita valittaessa täytyy ottaa huomioon se, mitä ne tarjoavat suhteessa siihen mitä halutaan luoda.

2.1.1 HTML5

HTML eli HyperText Markup Language on merkkäuskieli, jota on käytetty verkkosivujen luomiseen internetin luomisesta lähtien. HTML5 on siis 5. versio HTML-merkintäkielestä. Se sai W3C:n suositellun -statuksen 28.10.2014.

Vanhempiin versioihin verrattuna HTML5 tarjoaa uusia tajeja sekä elementtejä, eli palikoita joista verkkosivut rakentuvat. Versiomuutos myös muutti vanhentuneita ominaisuuksia joita ei enää suositella käytettäväksi.

Versio on otettu nopeasti käyttöön ja uusia ominaisuuksia tukevat kaikki suosituimmat selaimet.

Uusia ominaisuuksia ovat esimerkiksi uudet mediaelementit `<audio>` ja `<video>` -tagit jotka määrittävät ääni- ja videosisällön verkkosivulla. Nykyään yleinen ominaisuus on myös uusi HTML5 Geolocation API, jolla verkkosivu saa tiedon käyttäjän sijainnista. Tällä siis voidaan nopeuttaa erilaisia palveluita, joissa tieto käyttäjän nykyisestä sijainnista on tarpeellinen.

2.1.2 CSS3

CSS eli Cascading Style Sheets hallitsee verkkosivujen ulkonäköä. Selaimet määrittelevät eri elementtien oletustyyliä, mutta näitä voi muokata CSS-tiedoston avulla. Tiedoston avulla määritellään miltä verkkosivu näyttää käyttäjälle muuttamalla esimerkiksi fonttia, väriä ja kohteiden sijaintia sivulla.

Vanhempiin versioihin nähden CSS3 tarjoaa uusia työkaluja kohteiden animointiin ja muokkaukseen liittyen.

2.1.3 JavaScript

JavaScript on käyttäjänpuoleinen ohjelmointikieli, jonka avulla verkkosivulle voidaan lisätä dynaamisuutta. Käyttäjänpuoleisella viitataan siihen, kuinka se ei vaadi keskustelua palvelimen kanssa. Sen avulla voidaan poistaa, luoda ja muuttaa HTML-elementtejä verkkosivulla reaaliaikaisesti. Tämä siis mahdollistaa suuremman interaktion sivujen käyttäjän kanssa.

Toimiakseen JavaScript tarvitsee vain web-selaimen, joka tukee sitä. Laitteella siis ei ole merkitystä sen toimintaan. JS:ää käytetään yleisesti antamaan jonkinlaista palautetta käyttäjälle esimerkiksi virheilmoitusten muodossa. Se tarjoaa valmiita funktioita kuten `alert()` jolla näytetään käyttäjälle ilmoitusikkuna ja OK-nappi. Esimerkiksi `alert("Unohdit täyttää sähköposti-kentän!")` näyttää käyttäjälle ikkunan, jossa kyseinen viesti näkyisi.

2.1.4 jQuery

jQuery on yksi suosituimmista avoimen lähdekoodin JavaScript-kirjastoista. Sen tarkoituksena on nopeuttaa työskentelyä tekemällä toimintojen luomisesta helpompaa ja vähentää koodimäärää. jQueryn avulla pääsee helpommin käsiksi HTML-dokumenttien eri elementteihin ja tapahtumien käsittelyyn.

Pitää kuitenkin pitää mielessä, että jQuery on tehty JavaScriptillä. Se on vain optimoitu tekemään yleisiä funktioita tehokkaasti.

2.1.5 PHP

PHP on palvelinpuolen skriptikieli eli koodi suoritetaan serverillä josta tulkattu koodi lähetetään selaimelle. Käyttäjä siis näkee ainoastaan koodin generoiman HTML-lopputuloksen, mutta ei itse koodia. Sitä käytetään yleisesti esimerkiksi tietokantojen tiedonkäsittelyyn tai muuhun vastaavaan palvelinpuolen toimintaan.

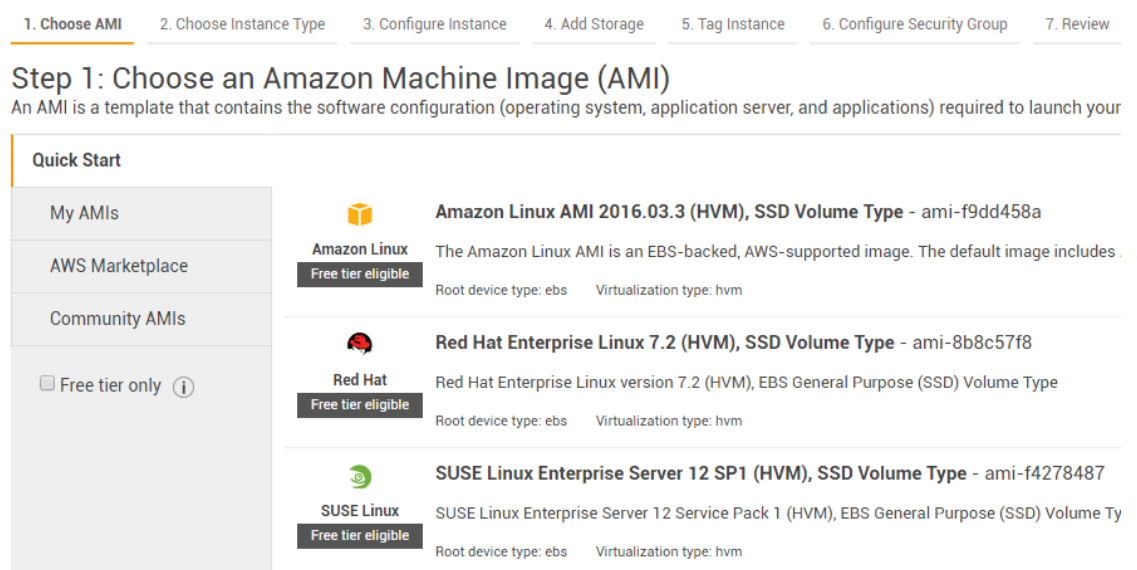
Koodia voi sisällyttää HTML-tiedoston sisälle josta se suoritetaan, mutta tässä tapauksessa palvelin täytyy laittaa suorittamaan HTML-tiedostot PHP:ssa. Helpompi tapa on nimetä tiedosto .php-päätteellä ja tehdä siitä PHP-tiedosto.

2.2 Amazon EC2

Amazon Elastic Compute Cloud eli Amazon EC2 on osa Amazonin tarjoamaa palvelukokonaisuutta. Se on verkkopalvelu joka tarjoaa skaalautuvaa laskentakapasiteettia AWS-pilvessä. Sen käyttäminen poistaa tarpeen investoida fyysisiin laitteisiin, jolloin käyttöönotto ja huolto helpottuu huomattavasti. Virtuaalipalvelimia voidaan helposti skaalata tarpeen mukaan antaen palvelimille lisää kapasiteettia.

2.2.1 EC2-instanssin luominen

Instanssien luominen on tehty erittäin helpoksi ja ymmärrettäväksi prosessiksi. Aloitetaan valitsemalla AMI eli Amazon Machine Image. AMI:a voi yksinkertaistettuna pitää virtuaalikoneen pohjapiirroksena, jonka pohjalta se luodaan. AWS tarjoaa oletuksena monia eri pohjia (Kuva 1), mutta lisäksi yhtiöiden ja käyttäjien tekemiä pohjia löytää Amazonin omasta kauppa-
paikasta.



Kuva 1 Aloitusruuutu näyttää erilaisia pohjia

Seuraavaksi valitaan instanssin tyyppi, jolla määritetään sille allokoituidut resurssit. Tärkeimmät tiedot ovat virtuaaliprosessorien ja keskusmuistin määrä, joiden tarve riippuu siitä millaista tehoa ollaan tavoittelemassa. Tämän projektin tapauksessa halvin t1.micro-instanssi riittää palvelimen pyörittämiseen sujuvasti. Instanssin tyyppi vaikuttaa sen tuntihinnoitteluun, joten tehtävään sopivan instanssin (Kuva 2) valitseminen on tärkeää.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types All generations Show/Hide Columns

Currently selected: t1.micro (Variable ECUs, 1 vCPUs, 0.613 GiB memory, EBS only)
 Note: The vendor recommends using a m1.small instance (or larger) for the best experience with this product.

	Family	Type	vCPUs	Memory (GiB)
<input checked="" type="checkbox"/>	Micro instances	t1.micro Free tier eligible	1	0.613
<input type="checkbox"/>	General purpose	t2.nano	1	0.5
<input type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1
<input type="checkbox"/>	General purpose	t2.small	1	2

Kuva 2 Instanssin tyyppin valinta

Kolmantena askeleena konfiguroidaan instanssin tarkempia tietoja. Tässä näkymässä voidaan määrittää IP-osoitteeseen liittyvät asiat, monitorointi, skaalaus ja palvelimen jakaminen muiden AWS-asiakkaiden kanssa. Tässä tapauksessa muutoksia ei tarvita, mutta automaattisen skaalauksen määrittäminen on hyvä pitää mielessä.

Neljäs askel on tallennustilan määrittäminen instanssille. Kaikkien instanssien tiedot tallennetaan EBS volyymin, eli virtuaaliselle kovalevyille. Valitaan siis volyymin koko sekä tyyppi, jolla määritellään volyymin nopeus IOPS-mittayksiköllä mitattuna. Esimerkin (Kuva 3) volyymi tarjoaa 100 IOPS tasaista nopeutta, mutta voi tilapäisesti nostaa sen maksimissaan 3000 IOPS nopeuteen. Haluttaessa voidaan myös salata kovalevyn sisältö.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Tag Instance 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-6a91b49a	10	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Kuva 3 EBS volyymin konfigurointi

Viimeinen olennainen asetus on instanssin suojausryhmän määrittäminen. Se on ryhmä palomuurin sääntöjä, joilla määritellään sallittu liikenne palvelimelle. Esimerkissä (Kuva 4) palomuri on säädetty sallimaan liikenne kolmeen eri porttiin kaikista IP-osoitteista. Voidaan siis helposti tehdä eri ryhmät julkaisu- ja kehitysversioille, jolloin kehitysversioon pääsee käsiksi vain sallituista osoitteista.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: Create a new security group

Select an existing security group

Security group name: LAMP powered by Bitnami-5-6-24-0-r17 on Ubuntu 14-04-3-AutogenByAWSM

Description: This security group was generated by AWS Marketplace and is based on recomr

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere 0.0.0.0/0
HTTP	TCP	80	Anywhere 0.0.0.0/0
HTTPS	TCP	443	Anywhere 0.0.0.0/0

Add Rule

Kuva 4 Palomuurin konfigurointi

Instanssi on nyt luotu ja käynnistetty, joten jäljellä on enää yhteyden muodostaminen. Tätä varten luodaan SSH-avainpari, jolla varmistetaan turvallinen yhteys palvelimelle. Tämä tuottaa ladattavaksi .pem-päätteisen tiedoston, joka on yksityinen avain käytettäväksi ssh-asiakasohjelmalla. Tämän jälkeen voidaan hallita luotuja instansseja (Kuva 5) jotka näkyvät listattuna hallintakonsolissa.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm State
Fore...	i-8faf7902	t1.micro	eu-west-1b	terminated		None
	i-d1400091	t1.micro	eu-west-1a	running	2/2 checks passed	None

Kuva 5 Lista instansseista EC2 Management Console -näkyvässä

2.3 LAMP-palvelin

LAMP on lyhenne sanoista Linux, Apache, MySQL ja PHP. Sillä kuvataan yleistä palvelimen kokoonpanoa, jossa palvelinkoneen käyttöjärjestelmänä on Linux, palvelinohjelmistona Apache, tietokantana MySQL ja palvelinpuolen skriptikielenä PHP (Linux Wiki, 2014). Tämä kokonaisuus valittiin projektin alustaksi sen joustavuuden sekä tekijän kokemuksen vuoksi.

Projektissa otettiin huomioon palvelimen turvallisuuden takaaminen ja haettiin AWS Marketplace -kaupasta ilmainen AMI. Palvelin olisi voitu pystyttää valitsemalla verrattain tyhjä pohja johon voisi käsin lisätä palvelinohjelmiston ja muita tarvittavia komponentteja. Projektissa kuitenkin oltiin luomassa julkisesti näkyvää verkkosivua, joten palvelimen tietoturvan olisi oltava kunnossa. Tässä tapauksessa riski turva-aukon luomisesta huonon konfiguraation seurauksena olisi huomattava verrattuna luotetun avoimen lähdekoodin yhteisön luomaan pohjaan.

Valinnassa päädyttiin ”LAMP Stack powered by Bitnami” pohjaan, mikä tarjoaa valmiiksi konfiguroituna kaiken projektiin tarvittavan samalla huolehtien turvallisuudesta. Heti asennuksen jälkeen palvelimelle ei lisätty

muuta paitsi Fail2ban tietoturvaohjelma, jonka avulla voidaan suojautua brute force -hyökkäyksiä vastaan. Ohjelma seuraa palvelimen lokeja ja estää yhteyden osoitteista jotka yrittävät arvata kirjautumistunnuksia tai etsiä haavoittuvuuksia. Ohjelma asennettiin lähinnä rajoittamaan saapuvia yhteyksiä, sillä palvelimeen pääsee käsiksi vain privaatin SSH-avainparin avulla, joka luotiin instanssin yhteydessä.

2.3.1 Linux

Bitnamin luomassa LAMP-palvelimessa käyttöjärjestelmänä toimii 64-bittinen Ubuntu 14.04.3. Bitnami on huolehtinut valmiiksi järjestelmän kovettamiseen liittyvistä asioista kuten sulkemalla kaikki portit paitsi 22 (SSH), 80 (HTTP) ja 443 (HTTPS). Tietokantaan ja sen hallintaohjelmaan pääsy on rajoitettu lokaaleihin yhteyksiin ja sovelluskohtaisia turvallisuusohjeita on noudatettu (Bitnami. 2016). Ainoa lisäys valmiiseen konfiguraatioon on fail2ban joka skannaa lokitiedostoja ja estää liian monta kirjautumisyritystä tekevät ip-osoitteet ottamasta yhteyttä palvelimeen.

```
sshd[15895]: input_userauth_request: invalid user to [preauth]
sshd[15897]: input_userauth_request: invalid user christian [preauth]
sshd[15945]: input_userauth_request: invalid user nsroot [preauth]
sshd[15994]: input_userauth_request: invalid user asher [preauth]
sshd[16051]: input_userauth_request: invalid user luke [preauth]
sshd[16053]: input_userauth_request: invalid user deploy [preauth]
sshd[16135]: input_userauth_request: invalid user anonymous [preauth]
sshd[16240]: input_userauth_request: invalid user admin [preauth]
sshd[16790]: input_userauth_request: invalid user sysadm [preauth]
sshd[16834]: input_userauth_request: invalid user nologin [preauth]
sshd[17930]: input_userauth_request: invalid user kevin [preauth]
sshd[17977]: input_userauth_request: invalid user jonathan [preauth]
sshd[18213]: input_userauth_request: invalid user leo [preauth]
sshd[18416]: input_userauth_request: invalid user nsrecover [preauth]
sshd[18598]: input_userauth_request: invalid user awt [preauth]
sshd[18600]: input_userauth_request: invalid user brody [preauth]
sshd[18714]: input_userauth_request: invalid user support [preauth]
sshd[18758]: input_userauth_request: invalid user support [preauth]
sshd[18798]: input_userauth_request: invalid user support [preauth]
sshd[18842]: input_userauth_request: invalid user support [preauth]
```

Kuva 6 Kirjautumisyrityksiä 10 tunnin aikaväliltä

Katsottaessa palvelimen lokitiedostoja komennolla

```
" tail /var/log/auth.log | grep 'sshd' "
```

paljastui ikävä puoli julkisen palvelimen ylläpitämisestä. Tiedosto sisälsi satoja automaattisia yrityksiä kirjautua sisään ssh-yhteydellä käyttäen yleisiä tunnuksia (Kuva 6). Listalta löytyy erilaisten palveluiden oletuskäyttäjänimiä joilla botit yrittävät päästä sisään palvelimiin joissa ei ole muutettu asennuksen jälkeisiä oletuskäyttäjätunnuksia ja salasanoja. Tästä lähti idea asentaa palvelimelle fail2ban joka muokkaa automaattisesti palvelimen palomuurin konfiguraatiota estämään ip-osoitteet jotka ovat lähettäneet liikaa epäonnistuneita pyyntöjä.

2.3.2 Apache

Apache HTTP Server on HTTP-palvelinohjelma, joka perustuu avoimeen lähdekoodiin. Projekti käynnistettiin vuonna 1995, ja on yksi suosituim-

mista palvelinohjelmista tähän päivään saakka. Sen markkinaosuus aktiivisten verkkosivujen joukossa on melkein 50 prosenttia (Netcraft 2016).

```
daemon 6566 0.0 3.8 1307984 23488 ? S1 Oct15 0:46 /opt/bitnami/apache2/b
in/httpd -f /opt/bitnami/apache2/conf/httpd.conf
root 12095 0.0 1.3 200596 8220 ? Ss Sep21 1:49 /opt/bitnami/apache2/b
in/httpd -f /opt/bitnami/apache2/conf/httpd.conf
daemon 12100 0.0 4.4 1311480 27028 ? S1 Sep21 9:38 /opt/bitnami/apache2/b
in/httpd -f /opt/bitnami/apache2/conf/httpd.conf
daemon 12101 0.0 4.2 1308216 25748 ? S1 Sep21 9:33 /opt/bitnami/apache2/b
in/httpd -f /opt/bitnami/apache2/conf/httpd.conf
```

Kuva 7 Apache pyörii daemon käyttäjän alla

Bitnami lupaa noudattaa palveluiden turvallisuusohjeita, joten tarkistetaan pitääkö väite paikkaansa pienellä testillä. Tarkistetaan että Apache -palvelu pyörii käyttäjällä, jolla on rajoitetut oikeudet palvelimella käyttämällä komentoa

```
" ps aux | egrep '(apache|httpd)' "
```

joka näyttää minkä käyttäjän oikeuksilla apache on käynnissä. Käyttäjänä oli daemon joka on perinteinen nimi rajattujen oikeuksien käyttäjälle tai ryhmälle (Kuva 7). Mikäli Apache olisi ollut käynnissä esimerkiksi root-käyttäjän oikeuksilla, kyseessä olisi ollut tietoturvariski.

3 KÄYTTÄJIEN HALLINTAJÄRJESTELMÄ

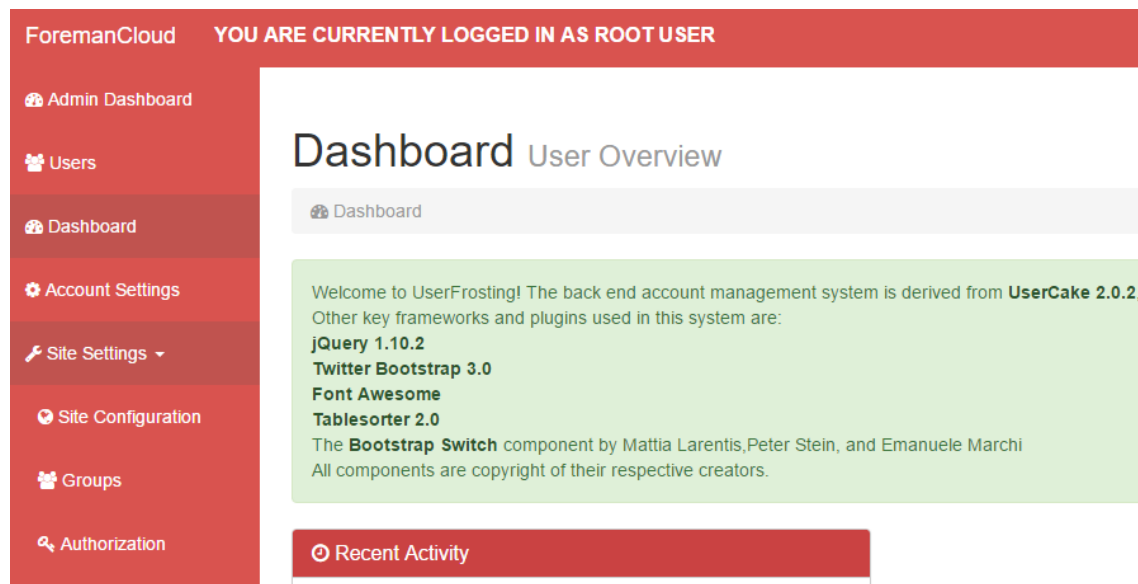
Sovellusta määrittäessä kävi selväksi että ainakin tulevaisuutta varten tarvitaan käyttäjänhallintajärjestelmä. Backendiin sopivista kielistä allekirjoittaneella oli eniten kokemusta PHP:sta, joten tuntui luonnolliselta aloittaa etsiminen tähän pohjautuvista järjestelmistä.

Userfrosting on PHP:hen pohjautuva käyttäjänhallintajärjestelmä. Sen avulla pääsee nopeasti alkuun käyttäjähallintajärjestelmän luomisessa, sillä frontend ja backend koodi ovat erillään toisistaan ja kommunikoivat AJAXin ja JSONin välityksellä. Järjestelmä vaikutti helposti käytettävältä ja muokattavalta, sekä ratkaisi hankalia tietoturvaan liittyviä ongelmia.

3.1 Käyttöönotto

Järjestelmän käyttöönotto on tehty hyvin sulavaksi kokemukseksi, samaan tyyliin kuin suosittu WordPress-alusta. Käyttäjän tarvitsee vain sijoittaa ladatut tiedostot WWW-palvelimen kotikansioon ja navigoida selaimella osoitteeseen ”palvelimen-URL/install”. Tämän jälkeen tarvitsee vain seurata järjestelmän antamia ohjeita admin-käyttäjätunnusten luomiseksi.

Installeri siis pystyttää SQL-tietokannan ja hallinointiin tarvittavan käyttäjätunnuksen, minkä jälkeen uusia käyttäjiä ja muita säätöjä voidaan alkaa tekemään Bootstrapin pohjalta luodussa käyttöliittymässä. Heti asennuksen jälkeen systeemissä on oletuksena muutamia tärkeitä sivuja navigaatiopalkkiin liitettynä (Kuva 8).



Kuva 8 Asennuksen jälkeinen oletusnäkyvä ja valmiit sivut navigaatiopalkissa

Aluksi käydään muokkaamassa sivuston asetuksia ”Site Configuration” – sivulla (Kuva 9). Tärkeimpänä muutoksena on uusien käyttäjien rekisteröinnin estäminen, ja varmistaa että ”Admin Root URL” on oikea osoite. Tässä tapauksessa halutaan varmistaa että vain halutut käyttäjät pääsevät palveluun käsiksi. Uudet käyttäjät siis luodaan manuaalisesti admin-tunnuksilla. Samassa paneelissa voidaan myös hallinnoida sähköposti aktivoinnin määrittystä, sivun kieltä ja muita hallinnollisia asioita.

Site Settings

The image shows a configuration page titled 'Configuration'. It contains several settings: 'Site Name' is set to 'ForemanCloud'; 'Admin Root URL' is set to 'esimerkkiosite.com/'; 'Account Management Email' is set to 'empty@empty.com'; 'User Registration' is a toggle switch set to 'OFF' with the text 'Specify whether users can create new accounts by themselves.' below it; and 'Email Login' is a toggle switch set to 'OFF' with the text 'Specify whether users can login via email address or username instead of just username.' below it.

Kuva 9 Sivuston asetukset

Lisäksi käydään ”Authorization”-sivulla (Kuva 10) varmistamassa että kaikki sivut paitsi sisäänkirjautumis- ja virhesivusto ovat näkyvissä vain sisäänkirjaantuneille käyttäjille. Kaikki sivut joissa ei ole ”Private” -ruksia näkyvät kenelle tahansa joka tietää oikean osoitteen, sillä niissä ei ylläpi-

detä käyttäjätarkistusta. Samalla sivulla voidaan merkata sivustoja eri käyttäjäryhmien käytettäväksi, mikäli halutaan tehdä esimerkiksi vierasryhmä jolla on rajatut oikeudet palvelun käyttöön.

Page-level authorization

Page ID	Filename	Private	User	Administrator
1	forms/table_users.php	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	account/logout.php	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	account/dashboard.php	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	account/dashboard_admin.php	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	account/account_settings.php	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	account/site_authorization.php	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Kuva 10 Palvelun sivujen valtuutus eri käyttäjäryhmille

3.2 Uuden käyttäjän luominen

Kun sivusto on kunnossa edellisten toimien jälkeen, voidaan testata käyttäjänhallintajärjestelmän toimintoja. Luodaan uusi käyttäjä, jolla on oikeudet sivuston hallinointiin. Siirrytään ”Users”-sivulle (Kuva 11) jossa voidaan tarkastella kaikkia palveluun liitettyjä käyttäjiä. Tällä sivulla tunnuksia voidaan muokata, poistaa käytöstä tai poistaa kokonaan.

User/Info	Registered Since	Last Sign-in	Status/Actions
Admin Account <i>Master Account</i> ✉		Sunday Sep 25th, 2016 4:25 pm	Active
Esimerkki Matti (mattinen) <i>Rakennusvalvoja</i> ✉ matti@email.com	Wednesday Sep 21st, 2016 4:53 pm	Wednesday Sep 21st, 2016 4:53 pm	Active

<< < 1 - 2 / 2 (2) > >>

Jump to Page: • Show:

[+ Create New User](#)
[View All Users](#)

Kuva 11 Käyttäjätunnusten hallinointi

Sivulta löytyy ”Create New User” -nappi uusien käyttäjien luomiseksi. Painamalla tätä nappia avautuu paneeli (Kuva 12), johon uuden käyttäjän tiedot täytetään. Oleellisin tieto on käyttäjän ”Groups”-kenttä, jossa määritetään mihin ryhmiin käyttäjä kuuluu sekä heidän pääryhmänsä. Ryhmät määrittävät käyttäjälle sallitut toiminnot ja sivut, pääryhmä määrittää mm.

minkä ryhmän tyyli tiedostoa käytetään palvelun näyttämiseen sekä oletus-sivun, johon käyttäjä päätyy kirjaututtuaan sisään.

Kuva 12 Uuden käyttäjän luominen

3.3 Uuden sivun liittäminen navigaatiopalkkiin

UserFrosting ei sisällä käyttöliittymää navigaatiopalkin sisällön muokkaamiseen, joten muutokset täytyy tehdä suoraan tietokantaan. Avataan MySQL userfrosting tietokannan sisältä löytyvä ”uf_nav”-taulu (Kuva 13), johon liitetään tietoja uudesta sivusta. Menu-kolumnissa valitaan sivun navigointipalkki jossa sivun painike sijaitsee, tässä tapauksessa ”left” mikä tarkoittaa vasemmalla sijaitsevaa päävalikkoa. Page-kolumni osoittaa uuden sivun osoitteen sivuston pääosoitteeseen liitettynä. Name-kolumni määrittää uuden sivun nimen. Position-kolumni määrittelee napin sijainnin navigaatiopalkissa, pienin numero näytetään ylimpänä painikkeena. Class-name-kolumni on uniikki tunniste jonka avulla userfrosting tunnistaa sivun sekä määrittää mikä sivu on avattuna. Icon-kolumni sisältää CSS ikoni-luokan jota käytetään navigaatiopalkissa. Parent_id-kolumni sisältää ala-nappien tilanteessa sen napin tunnisteiden jonka alta ne avautuvat. Kyseessä on yksinkertainen nappi joka ei avaudu, joten arvoksi laiteetaan nolla.

			id	menu	page	name	position	class_name	icon	parent_id				
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	1	left	account/dashboard_admin.php	Admin Dashboard	1	dashboard-admin	fa fa-dashboard	0
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	2	left	account/users.php	Users	2	users	fa fa-users	0
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	3	left	account/dashboard.php	Dashboard	3	dashboard	fa fa-dashboard	0
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	4	left	account/account_settings.php	Account Settings	4	settings	fa fa-gear	0
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	5	left-sub	#	Site Settings	5		fa fa-wrench	0
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	6	left-sub	account/site_settings.php	Site Configuration	6	site-settings	fa fa-globe	5
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	7	left-sub	account/groups.php	Groups	7	groups	fa fa-users	5
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	8	left-sub	account/site_authorization.php	Authorization	8	site-pages	fa fa-key	5
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	9	top-main-sub	#	#USERNAME#	1	site-settings	fa fa-user	0
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	10	top-main-sub	account/account_settings.php	Account Settings	1		fa fa-gear	9
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	11	top-main-sub	account/logout.php	Log Out	2		fa fa-power-off	9
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	12	left	account/tyomaa.php	Tyomaa	10	tyomaa-tarkastelu	fa fa-database	0

Kuva 13 Uf_nav taulun sisältö, alimpana uuden sivun tiedot

Tämän jälkeen uusi sivu täytyy liittää manuaalisesti vähintään yhteen ryhmään. Avataan uf_nav_group_matches taulu (Kuva 14) samasta tietokannasta ja liitetään se ”Administrator” ryhmään. Taulusta löytyy menu_id ja group_id kolumnit. Nämä vastaavat ”uf_nav” sekä ”uf_groups” tauluista löytävää id-numeroa.

				id	menu_id	group_id			
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	12	12	1
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	11	8	2
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	10	7	2
<input type="checkbox"/>		Muokkaa		Kopioi		Poista	9	6	2

Kuva 14 Uusi sivu (id 12) liitettynä Administrator-ryhmään (id 1)

Uuden sivun tiedostoon täytyy myös liittää koodinpätkä, jonka avulla järjestelmä tietää mikä sivu on auki. Funktiolle ”renderMenu” täytyy siis syöttää aiemmin tietokantaan luotu class_name ”tyomaa-tarkastelu”.

```
<?php
echo renderAccountPageHeader(array("#SITE_ROOT#" => SITE_ROOT, "#SITE_TITLE#" => SITE_TITLE, "#PAGE_TITLE#" => "Tyomaa"));
echo renderMenu("tyomaa-tarkastelu");
?>
```

Uusi sivu on nyt luotu ja liitetty järjestelmän navigaatiopalkkiin. Lopuksi täytyy käydä ”Authorization”-sivulla ruksaamassa ”Private”-valinta varmistuen sivun näkyvyys vain halutuille käyttäjille.

3.4 Turvallisuus

UserFrosting on suunniteltu vastaamaan yleisimpiin ongelmiin mitä verkkosivujen kovettamisessa tulee vastaan. Tämä on tärkeää, sillä AWS-ympäristössä asiakas on vastuussa palvelimensa tietoturvallisuudesta. Amazon huolehtii vain omien tilojensa, laitteiden fyysisen, verkkoinfrastruktuurin ja virtualisointi-infrastruktuurin turvallisuudesta (Dobodorov, Yinal Ozkan. 2013, 5).

3.4.1 Salasanojen turvaaminen

Yksi pahimpia virheitä salasanojen suhteen on tallentaa ne tietokantaan sellaisenaan ilman muutoksia. Tässä tapauksessa mikäli tietokantaan pääsee ulkopuolinen käsiksi, käyttäjien tiedot ovat heti käytettävissä. Tämä voidaan estää käyttämällä hashaus algoritmia käyttäjien salasanoihin ennen niiden tietokantaan tallentamista. Näin tehdään hyökkäjälle epätodennäköiseksi määrittää alkuperäinen salasana.

PHP:n versio 5.5 ja sitä ylemmät sisältävät funktiot `password_hash` ja `password_verify`. UserFrosting käyttää näitä palvelun salasanojen hashauksessa ja validoinnissa. `password_hash` käyttää oletuksena `bcrypt` algoritmia jota ei kirjoitushetkellä ole vielä murrettu. Tämän lisäksi UserFrosting lisää salasanan perään suolauksen estämään sanakirjahyökkäykset. Suolauksella tarkoitetaan satunnaista dataa joka sisällytetään hashaus prosessiin. Tämä estää mahdollisuuden murtaa salasana katsomalla listasta esilaskettuja syöte ja hash pareja (The PHP Group, 2016).

3.4.2 Cross-site request forgery (CSRF)

CSRF on hyökkäys, joka huijaa käyttäjän lähettämään vahingollisen pyynnön palvelimelle. Toimiakseen käyttäjän täytyy olla jo kirjautuneena sisään palveluun, jotta käyttäjällä olisi aktiivinen sessio verkkosivulla. Palvelin siis havaitsee session avulla että käyttäjä on todennettu, ja hyväksyy haitallista dataa tältä.

Puolustautuakseen tältä UserFrosting generoi käyttäjän kirjautuessa sisään uuden, satunnaisen tokenin. Token liitetään formeihin piilotettuna fieldinä, jonka palvelin tarkistaa ennen kuin se tekee mitään.

3.4.3 Cross-site scripting (XSS)

Tämä hyökkäys huijaa käyttäjän suorittamaan haitallisia skriptejä. Tämä on yleensä seurauksena käyttäjän syötteen tarkistuksen puuttumisesta. Käyttäjä siis pystyy lisäämään ylimääräisiä merkkejä kuten JavaScript/HTML tageja prosessoitavaksi palvelimelle, josta skripti päättyy toisen käyttäjän suoritettavaksi. Tämän vuoksi kaikki mahdolliset käyttäjän syötteet on sanitoitu, eli ne puhdistetaan ylimääräisistä merkeistä.

3.4.4 SQL-injektio

SQL-injektio huijaa käyttäjän sijaan palvelimen suorittamaan haitallista koodia SQL-lauseen muodossa. Suoritettu lauseke voi esimerkiksi poistaa SQL-tietokannan sisällön kokonaisuudessaan. Hyökkäys on mahdollinen, mikäli käyttäjä pystyy lisäämään SQL-kyselyyn oman lausekkeensa. Tältä voidaan välttyä parametrisoimalla SQL-kyselyt. Parametrisoidussa SQL-kyselyssä SQL-lause muodostetaan liittämällä käyttäjän syöttämät parametrit ennaltamäärätyihin paikkoihin.

4 PILVIPALVELUN TOTEUTUS

Pilvipalvelusta toteutettiin toiminnallinen prototyyppi suunnitelman pohjalta. Prototyyppi on käytettävissä AWS-palvelussa ylläpidetyssä palvelimessa. Prototyyppi ei sisällä kaikkia aloituskokouksessa suunniteltuja ominaisuuksia, mutta luo hyvän pohjan jatkolle. Projektista rajattiin aloituskokoukseen verrattuna pois tuotteiden CE-merkintätaulukko ja synkronointi tiedostopalveluihin. Prototyyppi kuitenkin sisältää sisäänkirjautumisjärjestelmän, työmaiden luonnin, kuvien ottamisen ja asiakirjojen muokkauksen sekä tulostamisen. Käyttökokemuksesta mobiililaitteilla on huolehdittu, sillä ne ovat pääasiainen käyttölaite työmailla paikan päällä oltaessa. Tietojen säilyvyys on varmistettu automaattisella varmuuskopiointilla.

4.1 Toiminnallisuus

Palvelun toiminta vaatii kommunikaatiota sivuston ja tietokannan välillä, jota varten palvelimelle ladattiin ”PHP-CRUD-API” REST-ohjelmointirajapinta. Rajapintaa voi siis vapaasti muokata ja käyttää omiin tarkoituksiin kohteesta riippumatta. REST tarkoittaa arkkitehtuurimallia jonka mukaan rajapinta on rakennettu. Avainasioita REST-mallissa ovat resurssien tunnistaminen uniikilla ID:llä ja resurssien relaatioiden linkitys (Abeysinghe 2008, 13).

Kyseessä on avoimen lähdekoodin rajapinta joka käyttää MIT-lisenssiä joten sen käyttö on sallittua kaupallisissa suljetun lähdekoodin ohjelmissa. Rajapinta pystyy toteuttamaan yleiset CRUD-operaatiot eli ”Create, Read, Update and Delete” funktiot. Näiden lisäksi rajapinta sisältää muita hyödyllisiä toimintoja joita käytetään palvelussa.

Rajapintaa käytetään palvelun toiminnallisuuden luomisessa hakemaan ja muuttamaan tietoa ilman että sivua tarvitsisi ladata uudelleen. Perinteisesti käyttäjät täyttävät lomakkeisiin tietoja ja painavat lähetyksen nappia joka lähettää tiedot palvelimelle käsiteltäväksi, minkä jälkeen sivu lataa itsensä uudelleen. Rajapinnan avulla voidaan rakentaa lomake joka päivittää tietyn kentän tietokannassa olevat tiedot heti muutoksen jälkeen, ilman erillistä lomakkeen lähetyksenä painallusta.

PHP-CRUD-API		
CRUD-operaatiot	SQL-komento	HTTP-metodi
Create (Luo)	INSERT	POST
Read (Hae)	SELECT	GET
Update (Muokkaa)	UPDATE	PUT
Delete (Poista)	DELETE	DELETE

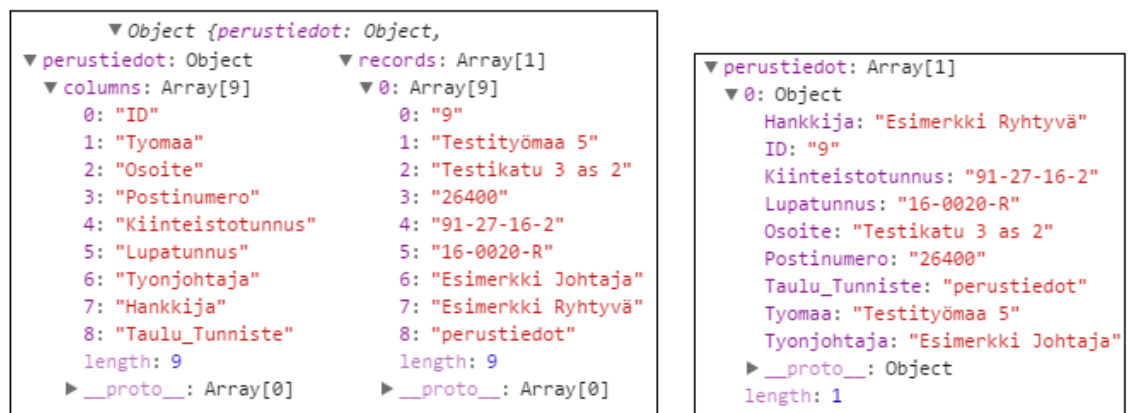
Kuva 15 PHP-CRUD-API käyttökaavio

Rajapintaa käytetään lähettämällä sille eri http-metodeilla kutsuja joita kaaviosta (Kuva 15) voidaan lukea. Jos halutaan esimerkiksi listata tietokannan taulusta tietyn tunnisteiden tiedot, lähetettäisiin rajapinnalle tämänlainen kutsu:

```
"GET http://osoite.com/api.php/perustiedot/9"
```

Rivistä voidaan lukea GET http-metodi, rajapinnan tiedostosijainti "api.php" sekä itse argumentit. Tämä hakisi perustiedot -taulusta kaikki tiedot riviltä, jonka id on numero yhdeksän.

Kaikki pyynnöt rajapinnalle lähetetään palvelussa AJAX-kutsuilla. Kutsut tehdään jQuery JavaScript-kirjaston merkinnöillä seuraavaan tapaan. Kuvassa näkyy kaksi eri tapaa toteuttaa kutsun palautus tällä rajapinnalla. Oikealla oleva kutsu sisältää rajapinnan mukana tulevan funktion joka muuntaa vastauksena tulevan objektin hierarkkiseen muotoon joka helpottaa sen käyttämistä. Nämä AJAX-kutsut palauttavat seuraavat objektit (Kuva 16):



Kuva 16 Vasemmalla muokkaamaton vastausena tuleva objekti ja oikealla hierarkkiseen muotoon muutettu objekti

Tuloksena saatuja objekteja voidaan käyttää tiedon esittämiseen sekä pohjana tietojen päivittämiseen. Tässä tapauksessa haetut tiedot syötetään Bootstrap-Table -nimiselle koodikirjastolle joka tuottaa niistä taulun. Tähän kirjastoon on lisätty vielä lisää toiminnallisuutta x-editable -lisäosalla joka sisällyttää tauluun mahdollisuuden muokata niitä reaaliajassa. X-editable laukaisee funktion, kun kenttää muokataan ja tallennetaan. Lisäämällä tähän funktioon oma rajapinta-kutsu voidaan tieto tallentaa heti kun muokkaus on tehty. Kyseessä on tiedon päivitys joten kutsu näyttää tältä:

```
" PUT http://osoite.com/api.php/perustiedot/2
{"Tyomaa":"Uusi nimi"} "
```

Kyseinen kutsu päivittäisi perustiedot -taulussa rivin jonka id on kaksi. Päivitettävä kolumni ja sen uusi arvo lähetetään JSON-muodossa.

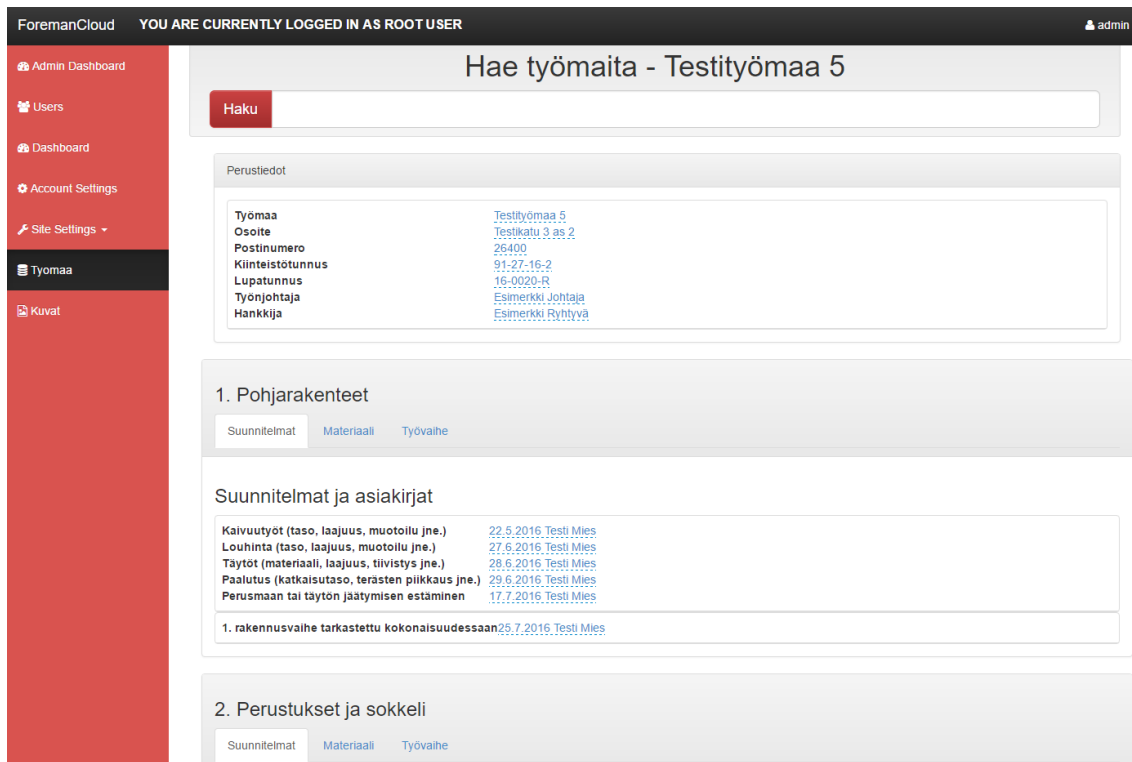
4.2 Käyttöliittymä

Palvelun käyttäjänhallinnan käyttöliittymä käyttää Bootstrap-koodikirjastoa muotoilun ulkoasun toteuttamiseen, joten samaa kirjastoa käytetään kaikkiin sivuun liitettyihin osiin pitäen ulkoasun samankaltaisena. Bootstrap on hyvin suosittu responsiivisten ja mobiiliystävällisten sivujen luomiseen. Projektissa käytetään versiota 3 sillä kirjaston uusi versio 4 ei ole vielä päässyt eteenpäin alpha-asteelta.

The screenshot shows the ForemanCloud admin interface. At the top, it says 'ForemanCloud YOU ARE CURRENTLY LOGGED IN AS ROOT USER' with a user profile icon for 'admin'. On the left is a red sidebar menu with options: Admin Dashboard, Users, Dashboard, Account Settings, Site Settings, Työmaa, and Kuvat. The main content area is split into two panels. The left panel, titled 'Uusimmat työmaat', displays a list of ten job entries: Testityömaa 5, Testimaa 50, Ankkalampi Kämpä, Meilammen Talo, Naapuruston Talo, Saarenmaan Omakotitaloalue, Katajan Päiväkoti, and Testimaa 1. The right panel, titled 'Luo uusi työmaa - Perustiedot', contains a form with fields for: Työmaan nimi, Osoite, Postinumero, Lupatunnus, Kiinteistötunnus, Rakennushankkeeseen ryhtyvä, and Työnjohtaja. A 'Valmis' button is located at the bottom of the form.

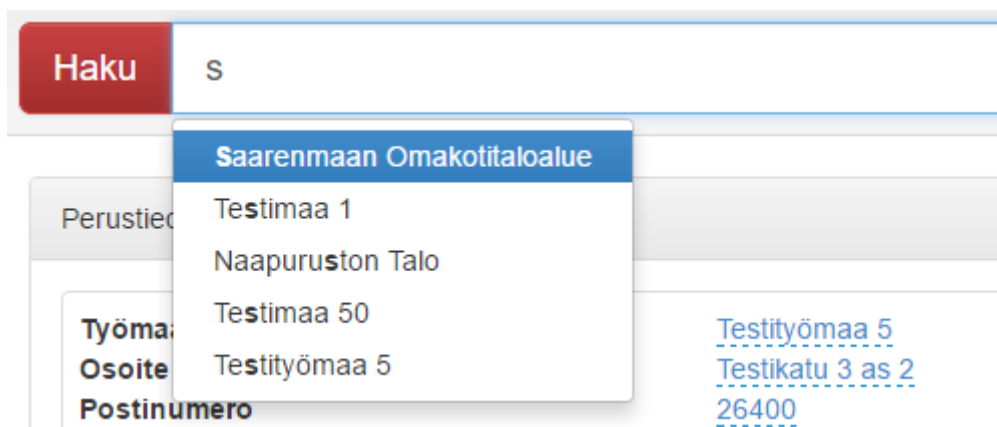
Kuva 17 Palvelun etusivu

Etusivu koostuu kahdesta Bootstrap-paneelistä (Kuva 17). Uusimmat työmaat hakee tietokannasta viimeiset 10 työmaata ja näyttää ne listassa. Klikkaamalla näitä avautuu kyseisen työmaan tiedot uudella sivulla (Kuva 18). Toinen paneeli sisältää lomakkeen jolla luodaan uusi työmaa.



Kuva 18 Työmaa-sivu

Työmaa-sivu näyttää työmaan perustiedot sekä täytettävät kohdat pientalotyömaan tarkastusasiakirjan mallista. Täytettävillä kohdilla on kolme eri aliotsikkoa täytettävänä joten ne jaettiin paneeleissa välilehtien taakse. Tämä toteutustapa pitää sivun pituuden kurissa ja helpottaa mobiilikäyttöä.



Kuva 19 Ennakoiva haku

Palvelun kaikki hakukentät sisältävät ennakoivan syötteen (Kuva 19) joka vertaa kirjoitettua tekstiä tietokannassa olevien työmaiden nimiin. Ominaisuus implementoitiin lisäämällä sivustolle Bootstrap 3 Typeahead -lisäosa, joka on yksittäinen JavaScript-tiedosto.

```

$("#typeahead").typeahead({
  source: function(query, process) {
    $.ajax({
      url: 'haku.php',
      type: 'POST',
      data: 'query=' + query,
      dataType: 'JSON',
      async: true,
      success: function(data) {
        process(data);
      }
    });
  }
});

```

Kuva 20 Typeahead jQuery funktio

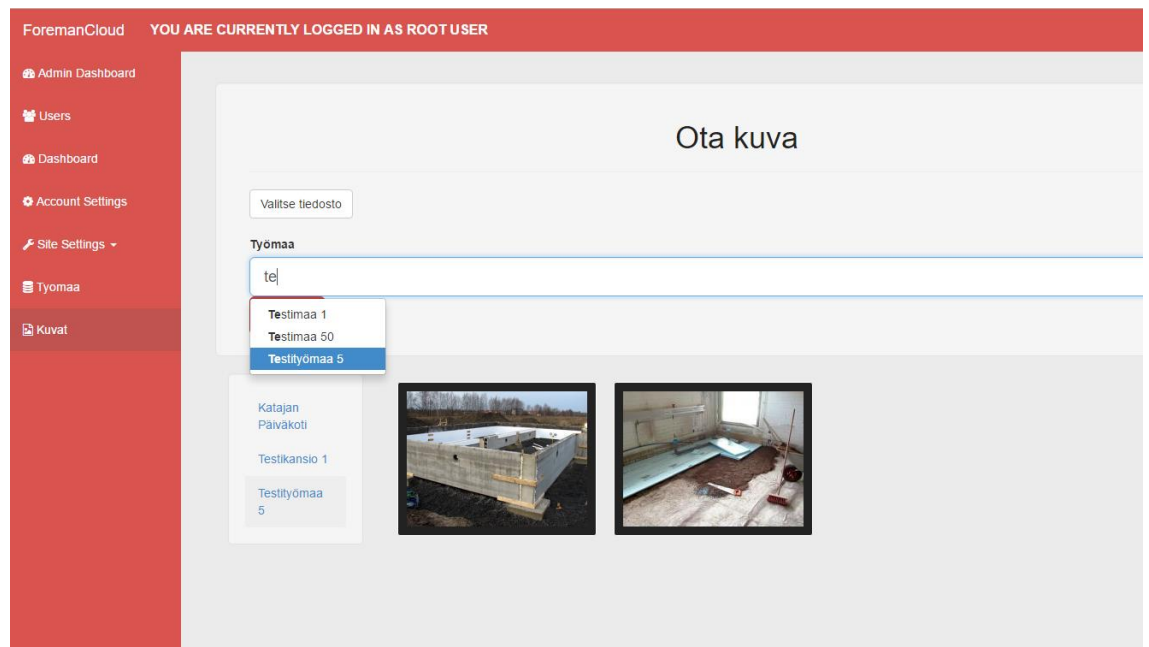
Funktio (Kuva 20) lähettää kirjoitetun tekstin ”haku.php”-tiedostolle joka sisältää sql-lausekkeen

```

`SELECT * FROM `perustiedot` WHERE `Tyomaa` LIKE
'${query}%'`

```

joka palauttaa saadut tulokset array-muodossa vastauksena sitä kutsuvalle funktiolle. Saatuaan vastauksen funktio prosessoi saadusta tiedosta kuvan 19 mukaisen ehdotuslistan nopeuttamaan työskentelyä.

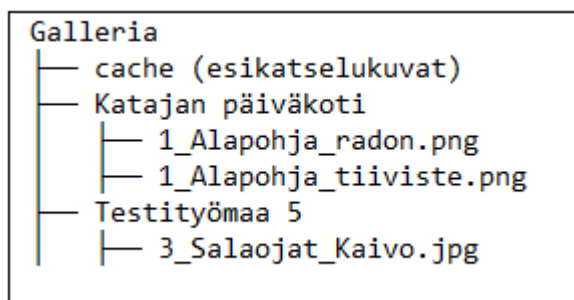


Kuva 21 Kuvien hallinta

Kuvat-sivu on yksinkertainen esitys palvelimella olevasta kansipohjaisesta rakenteesta (Kuva 21). Uuden kuvan liittäminen luo kansion työmaan nimellä, johon kuvat sijoitetaan. Valitse tiedosto -nappi avaa tietokoneella käytettynä normaalin tiedoston valinta dialogin ja mobiililla kamerasovelluksen josta kuvan voi lähettää suoraan valinnaksi.

4.3 Kuvien ottaminen

Valitut kuvat lähetetään ”upload.php” -tiedostolle käsiteltäväksi. Tiedosto varmentaa tiedoston tyypin ja tallentaa sen skaalattuaan sen ennalta määritettyyn kokoon. Näin säästetään tallennustilaa kun vältetään kännyköiden oletusasetuksilla otetuilta kuvilta jotka voivat olla hyvinkin suuria. Se ei korjaa hidasta lähetyssaikaa hitaalla yhteydellä mutta helpottaa kuvien katselua jälkikäteen. Kuvat tallennetaan kansiorakenteeseen josta ne myöhemmin luetaan (Kuva 22).



Kuva 22 Kuvien kansiorakenne

Kuvia katsellessa ensimmäistä kertaa niistä luodaan matalalaatuinen, pienikokoinen esikatselukuva joka tallennetaan palvelimelle erilliseen kansioon. Sivulla oleva koodi siis skannaa valitut kansiot ja niiden kuvat ja esittää ne mutta luo esikatselukuvan vain niille kuville joilla sellaista ei vielä ole.

4.4 Dokumenttien tulostus

Dokumentit tulostetaan käyttämällä valmista excel-pohjaa johon liitetään tiedot tietokannasta PHPEXCEL PHP-koodikirjaston avulla. Kirjaston avulla on mahdollista lukea ja kirjoittaa eri taulukkolaskentaohjelmien formaateihin (Kuva 23).

```

$objPHPExcel->getActiveSheet()
// 1. Pohjarakenne
->setCellValue('E17', '7.7.2016 Php Edit')
->setCellValue('H17', '7.8.2016 Php Edit')
->setCellValue('K17', '7.9.2016 Php Edit')
->setCellValue('E18', '7.7.2016 Php Edit')
->setCellValue('H18', '7.8.2016 Php Edit')
->setCellValue('K18', '7.9.2016 Php Edit')
;

```

RAKENNETEKNIKAN TARKASTUSASIAKIRJAN MALLI			
Rakennusvaihe:			
Rakennusvaiheen vastuuhenkilö varmentaa allekirjoituksellaan rakennusvaiheen tarkastetuksi, kun rakennustuotteiden kelpoisuus on osoitettu			
ja niiden tiedot koottu tuotekansioon sekä eri työvaiheet on tarkastettu			
Suunnitelmat ja asiakirjat:			
Vastaavan työnjohtajan tehtävänä on huolehtia siitä, että työmaalla on käytettävissä kuhunkin työvaiheeseen tarvittavat suunnitelmat, työselostukset, asennusohjeet ja muut työn toteutuksessa tarvittavat asiakirjat			
Materiaalin ja rakennustuotteen kelpoisuus:			
Rakennusvaiheen vastuuhenkilön tehtävä on todeta, että käytettävät materiaalit ja rakennustuotteet ovat suunnitelmien mukaisia ja niiden kelpoisuudesta on työmaalla riittävät selvitykset riittävine ohjeineen..			
Työvaiheen tarkastus:			
Työvaiheen tarkastuksen vastuuhenkilö merkitsee työvaiheen tarkastetuksi.			
Rakennusvaiheet ja niiden työvaiheet		Työvaiheiden eri osien tarkastukset, pvm:t ja allekirjoitukset	
1 Pohjarakenne		Suunnitelmat ja asiakirjat	Materiaalin ja rakennustuotteen kel
		Työvaiheen tarkastus	
Kaivutyöt (taso, laajuus, muotoilu jne.)	7.7.2016 Php Edit	7.8.2016 Php Edit	7.9.2016 Php Edit
Louhinta (taso, laajuus, muotoilu jne.)	7.7.2016 Php Edit	7.8.2016 Php Edit	7.9.2016 Php Edit
Täytöt (materiaali, laajuus, tiivistys jne.)			
Paalutus			
- katkaisutaso, terästen piikkaus jne.A22			
- työmaalla viranomaisleimalla varustettu			
paalutuksen tarkesuunnitelma			
Perusmaan tai täytön jäätyksen estäminen			
1. rakennusvaihe tarkastettu kokonaisuudessaan, pvm ja vastuuhenkilön allekirjoitus:			

Kuva 23 Malliesimerkki PHPEXCEL käytöstä

Kirjasto pystyy muokkaamaan useampaa taulukkoa yhdessä tiedostossa helpottaen dokumentin luomista suoraan tulostuskuntoon.

4.5 Varmuuskopiointi

Tietojen varmuuskopiointi toteutetaan AWS:stä löytyvällä EBS Snapshot -ominaisuudella. Se ottaa EBS-volyymistä kopion ja tallentaa sen Amazon S3 -palveluun kokonaisuudessaan. Ensimmäisen kopion jälkeen tapahtuvat kopiot ovat inkrementaalisia, eli tallennetaan ainoastaan muuttuneet tiedot tehden kopioinnista paljon nopeampaa. Volyymien kopiointi varmuuskopioi kaiken olennaisen, sillä tietokanta ja kuvat löytyvät samalta volyymiltä.

Step 1: Create rule

Create rules to automate actions in your AWS environment.

Event selector

Build a pattern that selects events for processing by your targets.

The screenshot shows the 'Schedule' configuration for an AWS Event rule. It features a dropdown menu labeled 'Schedule' with a downward arrow. Below it, there are two radio button options. The first is 'Fixed rate of' with a value of '24' and a unit dropdown set to 'Hours'. The second is 'Cron expression' with a value of '0 0 0 * * ?'. Below the cron expression field, there is a link that says 'Learn more about CloudWatch Events schedules.'

Kuva 24 Uuden säännön ajastus

Varmuuskopioinnin automatisointi onnistuu helposti AWS:n omilla työkaluilla. Avataan "CloudWatch"-konsoli ja avataan sivulta löytyvästä palkista "Events" josta löytyy "Create rule"-nappi. Tämä avaa näkymän jossa uusi sääntö voidaan luoda. Valitaan kuvan mukaisesti "Event selector" josta voidaan valita aikataulun luominen uudelle säännölle. Säännön voi ajastaa käynnistymään minuuttien, tuntien ja päivien välein. Tarkempia sääntöjä varten löytyy "Cron expression" jonka avulla säännön aktivointi voidaan säätää paljon tarkemmin esimerkiksi kerran kuukauteen (Kuva 24).

Targets

Select the targets to receive the events that match the rule you defined.

The screenshot shows the 'Targets' configuration for an AWS Event rule. It features a dropdown menu labeled 'Built-in target' with a downward arrow and a close button (X). Below it, there are two input fields: 'Action*' with the value 'Create a snapshot of an EBS volume' and 'Volume ID*' with the value 'vol-3f9ce638'. At the bottom, there is a button labeled '+ Add target*'.

Kuva 25 Säännön kohteen valinta

Ajastimen vierestä löytyy säännön kohteen valinta, joka määrittää aktivoituvan tapahtuman ja sen kohteen (Kuva 25). Samalla tavalla voidaan esimerkiksi ajastaa instanssin pysäytys varmuuskopioinnin ajaksi sekä sen uudelleenkäynnistyminen. Seuraavaksi annetaan säännölle nimi ja kuvaus sekä annetaan sisäänrakennetulle roolille oikeudet suorittaa tapahtuma.

Status	Name	Description
<div style="display: flex; align-items: center;"> <div style="margin-right: 10px;"> ● ↓ </div> <div> EBS_Varmuuskopiointi </div> </div>		Ota varmuuskopio joka päivä kohteen EBS volyymistä

Kuva 26 Uusi sääntö on aktiivinen

Tämän jälkeen uuden säännön voi tallentaa ja varmistaa sen toimivuuden palaamalla ”EBS Snapshots” listaan. Mikäli sääntö on ajastettu suoritettavaksi tietyn aikamäärään välein, se suoritetaan heti säännön luomisen jälkeen (Kuva 26). Tämän jälkeen se seuraa säädettyä ajastinta. Luotua kopiota suositellaan testattavaksi kuukausittain mahdollisten virheiden paljastamiseksi.

4.6 Laajennusmahdollisuudet

Projektilla on erityisen paljon mahdollisuuksia laajentua entistä isommaksi kokonaisuudeksi. Laajennusten mahdollisuudet vaihtelevat kohteittain ja vaatisivat joissain tapauksissa lähes tyhjästä uudelleen aloittamisen.

4.6.1 UserFrosting päivitys

Projektissa käytettiin versiota 0.2.2, mutta uusin versio 0.3.1 muuttaa kirjaston toimintatapaa radikaalisti. Uusi versio muuttaa koodin rakenteen täysin MVC-malliseksi, käyttää Composer -työkalua hallitsemaan eri kirjastoja sekä niiden riippuvuuksia ja muita moderneja komponentteja. Erot ovat kuitenkin niin suuria että projekti täytyisi rakentaa melkein tyhjästä uudelleen. Mikäli tähän päädyttäisiin kannatttaa harkita laajemman ja tunnetumman koodikirjaston ja sen lisäosien käyttöä.

4.6.2 Synkronointi tiedostopalveluun

Voitaisiin synkronoida projektin kuvatiedostot valittuun tiedostopalveluun kuten esimerkiksi Microsoft OneDriven käyttäen heidän tarjoamaa rajapintaa. Tämän avulla saataisiin lisätaso varmuuskopiointiin ja toinen tapa jakaa kuvia niille jotka niitä tarvitsevat. Samalla voitaisiin ehkä luoda varmuuskopio tietokannasta.

4.6.3 Sähköpostipalvelimen lisääminen

Sähköpostipalvelimen lisääminen projektiin sallisi lisää toiminnallisuutta kuten unohtuneen salasanan lähettämisen, sähköpostilla sisäänkirjautumisen ja esimerkiksi automaattisen raportoinnin lähetyksen tai vastaavaa.

4.6.4 LAMP-palvelimen päivitys

Voitaisiin päivittää uusimpaan AMI:n esimerkiksi uusimpaan Bitnamin luomaan PHP7 version sisältävään versioon. Yksinään PHP7 voisi parantaa palvelimen suorituskykyä huomattavasti.

5 YHTEENVETO

Palvelun suunnittelu ja toteuttaminen vei oletettua enemmän aikaa huonon kommunikaation takia, mutta lopputuloksena saatu prototyyppi on hyvä pohja miltä jatkaa kehitystä. Palvelun suurin toiminto eli asiakirjojen muokkaus on toteutettu modernin REST-rajapinnan avulla joten sen toimintojen laajentaminen tulevaisuudessa ei ole kovin vaikeaa. Muutamat asiat, jotka toimivat yksittäisenä kokonaisuutena, on luotu perinteisemmällä PHP-tekniikoilla mikä tekee koodin rakenteesta tarpeettoman sekavaa. Tämä oli kuitenkin odotettavissa, sillä projektissa ei sitouduttu MVC-malliin tai käytetty yhtenäistettyä PHP-rakennekirjastoa ja sen lisäosia.

Jälkikäteen katsottuna toteutus olisi voinut olla helpompaa sitoutumalla esimerkiksi Symfony- tai Laravel-koodikirjaston käyttöön. Nämä laajemat kirjastot ovat paremmin dokumentoituja sekä niille kehitetään enemmän lisäosia käytettäväksi. Kehityksessä olisi ollut hyvä myös käyttää jonkinlaista versionhallintaa kuten Git ja SVN, mikä tekisi koodin hallinnasta helpompaa varsinkin jatkokehitystä silmälläpitäen. Mobiiliulkoasua silmälläpitäen Bootstrap oli hyvä valinta tehtävään vähentäen tyylitiedosten säätämistä huomattavasti.

Toteutus AWS-palvelussa oli toimiva ja luotettava. Käytetty palvelin on ollut pystyssä lähes kaksi vuotta lähes täydellisellä ylläpitoajalla. Amazon ilmoittaa hyvissä ajoin suunnitelluista huoltokatkoista jotka ovat olleet ainoa syy palvelimen alasajoon. Toteutus saattaisi tulla halvemmaksi toteuttaa esimerkiksi DigitalOcean-palvelussa joka tarjoaa hyvin samankaltaisia toimintoja kuin AWS. Projektin teko opetti paljon uusia asioita varsinkin pilvipalveluihin liittyen ja oli kokonaisuutena positiivinen kokemus.

LÄHTEET

Amazon Web Services Inc. 2016. AWS Glossary. Viitattu 21.3.2016
<http://docs.aws.amazon.com/general/latest/gr/glos-chap.html>

Apache Software Foundation. 2016. Apache http Server Project. Viitattu 21.3.2016
<http://httpd.apache.org/>

Bitnami. Frequently Asked Questions For Bitnami Cloud Hosting. 2016. Viitattu 12.10.2016
<https://docs.bitnami.com/bch/faq/>

Chandan, Kumar. Geek Flare. Apache Web Server Hardening & Security Guide. 2016. Viitattu 16.10.2016
<https://geekflare.com/apache-web-server-hardening-security/>

Fail2ban Wiki. 2014. Main Page. Viitattu 21.3.2016
http://www.fail2ban.org/wiki/index.php/Main_Page

Jinesh, Varia. Amazon Web Services Inc. 2014. Tips for Securing Your EC2 Instance. Viitattu 26.4.2016
<http://aws.amazon.com/articles/1233/>

Linux Wiki. 2014. LAMP. Viitattu 21.3.2016
<https://www.linux.fi/wiki/LAMP>

Netcraft. April 2016 Web Server Survey. 2016. Viitattu 1.5.2016
<http://news.netcraft.com/archives/2016/04/21/april-2016-web-server-survey.html>

OWASP Foundation. Cross-Site Request Forgery (CSRF). 2015. Viitattu 1.5.2016
[https://www.owasp.org/index.php/Cross-Site_Request_Forgery_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF))

OWASP Foundation. Cross-Site Scripting (XSS). 2016. Viitattu 1.5.2016
[https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))

Samisa, Abeysinghe. *RESTful PHP Web Services*. 2008. Packt Publishing. Viitattu 17.10.2016

The PHP Group. Safe Password Hashing. 2016. Viitattu 26.9.2016
<http://php.net/manual/en/function.password-hash.php>

The PHP Group. 2016. What is PHP? Viitattu 18.1.2016
<http://php.net/manual/en/intro-what-is.php>

Todorov, Dob. Amazon Web Services Inc. 2013. AWS Security Best Practices. Viitattu 1.5.2016.

http://media.amazonwebservices.com/AWS_Security_Best_Practices.pdf

Userfrosting. 2016. About. Viitattu 13.2.2016

<http://www.userfrosting.com/0.2.2/about/>

Userfrosting. 2016. Getting Started with Userfrosting 0.2.0. Viitattu 25.4.2016

<https://github.com/userfrosting/UserFrosting/wiki/0.2.0-only-Getting-Started-with-UserFrosting>

Wilde, Bryan. jQuery vs JavaScript: What's the Difference Anyway? 2013. Viitattu 18.1.2016

<https://blog.udemy.com/jquery-vs-javascript/>

W3C. 2014. HTML5 W3C Recommendation 28 October 2014. Viitattu 18.1.2016

<https://www.w3.org/TR/2014/REC-html5-20141028/>

W3C. 2016. HTML5 New Elements. Viitattu 18.1.2016

http://www.w3schools.com/html/html5_new_elements.asp

W3C. 2016. HTML5 Geolocation. Viitattu 18.1.2016

http://www.w3schools.com/html/html5_geolocation.asp